Software Project Management

Comprehensive Lecture Notes

Based on: Bob Hughes & Mike Cotterell · PMBOK 5th Ed. · Robert K. Wysocki
BT Levels 2–4 | Domain: Cognitive | Credit Hours: 3(3,0)

Table of Contents

1. Introduction to Software Project Management
2. Project Management Concepts
3. Project Management Tools
4. PMI Knowledge Areas
5. PMI Framework
6. PMI Process Groups
7. Understanding Organizations
8. Project Planning
9. Project Evaluation
10. Selection of an Appropriate Approach
11. Software Effort Estimation
12. Activity Planning
13. Risk Management
14. Evaluating Risks to the Schedule
15. Risk Control
16. Configuration Management & Maintenance
17. Environment for Configuration Control
18. Resource Allocation
19. Monitoring & Control
20. Review and Evaluation
21. Challenges of Outsourcing in Project Management

1. Introduction to Software Project Management

Software Project Management (SPM) is the discipline of planning, organizing, controlling, and directing software development projects to achieve defined goals within constraints of scope, time, cost, and quality.

1.1 What is a Software Project?

• A project is a temporary endeavor with a defined start and end, producing a unique product, service, or result.

• Software projects are characterized by:

  – Intangibility — software cannot be physically seen or touched during development

  – Complexity — large number of interdependencies between components

  – Flexibility — requirements often change during execution

  – Novelty — most software projects involve some degree of new problem-solving

• Differs from hardware projects: no raw material costs, replication is near-zero cost, failure points are logical not physical.

1.2 The Project Triangle (Iron Triangle)

• Three competing constraints that every project manager must balance:

  – Scope — what work is included in the project

  – Time — the deadline or schedule

  – Cost — the budget and resource expenditure

• Changing one constraint forces adjustment in at least one other.

• Quality is sometimes added as a fourth constraint (diamond model).

Constraint	Increase Effect	Decrease Effect
Scope	More time & cost required	Faster & cheaper delivery
Time	Higher cost or reduced scope	Lower cost (if planned)
Cost	More resources, faster delivery	Reduced scope or delayed delivery

1.3 Software Project Failure — Common Causes

• Incomplete or changing requirements

• Lack of user involvement

• Poor project management

• Insufficient testing

• Technology incompetence

• Unrealistic time estimates (Hofstadter's Law: always takes longer than expected)

1.4 SPM vs General Project Management

Aspect	General PM	Software PM
Product	Physical deliverable	Intangible software
Measurement	Progress is visible	Progress is abstract

Change	Costly to change	Easy to change (but risky)
Testing	QA at end	Ongoing throughout SDLC

2. Project Management Concepts

Project Management is the application of knowledge, skills, tools, and techniques to project activities to meet project requirements (PMI definition).

2.1 Project Life Cycle

- All projects pass through phases from start to finish — collectively called the Project Life Cycle (PLC).
- Four standard phases:
  - Initiation — define the project at a high level; produce Project Charter
  - Planning — develop scope, schedule, cost, risk, and quality plans
  - Execution — coordinate resources and carry out the plan
  - Closure — formal acceptance, lessons learned, resource release
- Some models add a Monitoring & Controlling phase that runs concurrently with Execution.

2.2 Project vs Program vs Portfolio

Term	Definition	Example
Project	Single temporary endeavor	Develop a payroll module
Program	Group of related projects managed together	Entire ERP system rollout
Portfolio	Collection of programs/projects aligned to strategic goals	IT transformation strategy

2.3 Project Stakeholders

- Stakeholder — any individual or group affected by or affecting the project.
- Categories:
  - Internal — project team, project manager, functional managers, executives
  - External — clients, end users, vendors, government bodies, the public
- Stakeholder management steps:
  - Identify stakeholders
  - Analyze their interest and influence (Power-Interest Grid)
  - Plan engagement strategy
  - Manage and monitor engagement throughout the project

2.4 Project Charter

- A document that formally authorizes a project.
- Contents:
  - Project title and description
  - Project purpose and justification
  - Measurable objectives and success criteria
  - High-level requirements
  - High-level project description, boundaries, and key deliverables
  - Overall project risk
  - Summary milestone schedule
  - Pre-approved budget

  – Project manager authority level
  – Name and authority of project sponsor

2.5 Project Scope Management

- Scope defines the work required to deliver a product with specified features.
- Product Scope — features and functions of the product/service.
- Project Scope — work that must be performed to deliver the product.
- Scope Creep — uncontrolled expansion of scope without adjustment to schedule/cost.
- Work Breakdown Structure (WBS):
  – Hierarchical decomposition of total project scope
  – Lowest level element = Work Package
  – 100% Rule: WBS captures ALL work; no work exists outside the WBS

3. Project Management Tools

Project management tools are software applications and graphical techniques used to plan, schedule, track, and control project activities.

3.1 Gantt Chart

- Bar chart showing tasks on the vertical axis and time on the horizontal axis.
- Each task represented by a horizontal bar spanning its start and end dates.
- Pros:
  - Simple, easy to understand by non-technical stakeholders
  - Clearly shows project timeline and current status
- Cons:
  - Does not show task dependencies explicitly
  - Does not show resource loading
  - Becomes cluttered for large projects

3.2 Network Diagrams (PERT / CPM)

- PERT — Program Evaluation and Review Technique
  - Uses three time estimates: Optimistic (O), Pessimistic (P), Most Likely (M)
  - Expected time: $t_e = (O + 4M + P) / 6$
  - Handles uncertainty in task durations
- CPM — Critical Path Method
  - Uses single deterministic time estimate per task
  - Identifies the Critical Path — longest path through the network
  - Critical Path = sequence of tasks with zero float (slack)
- Key terms:
  - ES (Early Start): earliest a task can begin
  - EF (Early Finish): ES + Duration
  - LS (Late Start): latest a task can start without delaying the project
  - LF (Late Finish): LS + Duration
  - Float/Slack = LS – ES = LF – EF

3.3 Software Tools

Tool	Primary Use	Type
MS Project	Full-featured scheduling, resource management	Commercial
Jira	Agile tracking (Scrum/Kanban boards)	Commercial/Cloud
Trello	Kanban-style task boards	Commercial/Free tier
Asana	Task & project tracking	Commercial/Free tier
GanttProject	Open-source Gantt charting	Free
Primavera P6	Large-scale construction & engineering PM	Commercial

3.4 Other Graphical Tools

- Responsibility Assignment Matrix (RAM / RACI):
  - R = Responsible, A = Accountable, C = Consulted, I = Informed
  - Maps project roles against WBS elements
- Milestone Chart:
  - Shows only key events (milestones), not duration of tasks
  - Used for high-level executive reporting
- Earned Value Management (EVM) Charts:
  - Plots Planned Value (PV), Earned Value (EV), Actual Cost (AC) over time
  - Enables quantitative schedule and cost performance measurement

4. PMI's Knowledge Areas

PMI defines 10 Knowledge Areas in the PMBOK Guide — each representing a complete set of concepts, terms, and activities that make up a professional field of specialization within project management.

4.1 Integration Management

• Identifies, defines, combines, unifies, and coordinates processes and activities. Includes Project Charter, Project Management Plan, Direct & Manage Work, Monitor & Control, and Close Project.

4.2 Scope Management

• Ensures the project includes all the work required — and only the work required. Includes requirements collection, WBS creation, scope validation, and scope control.

4.3 Schedule Management

• Manages timely completion. Includes activity definition, sequencing, duration estimating, schedule development (CPM/PERT), and schedule control.

4.4 Cost Management

• Planning, estimating, budgeting, financing, funding, managing, and controlling costs. Includes cost estimating, budget determination, cost control (EVM).

4.5 Quality Management

• Determines, manages, and controls quality policies. Includes quality planning, quality assurance (process audits), and quality control (product inspection).

4.6 Resource Management

• Identifies, acquires, and manages the resources needed (team + physical). Includes team development, team management, and physical resource control.

4.7 Communications Management

• Ensures timely and appropriate planning, collection, creation, distribution, storage, retrieval, and disposition of project information.

4.8 Risk Management

• Increasing probability and impact of positive events; decreasing negative events. Includes risk identification, qualitative analysis, quantitative analysis, response planning, and monitoring.

4.9 Procurement Management

• Purchasing or acquiring products, services, or results from outside the project team. Includes make-or-buy decisions, contract types, vendor management.

4.10 Stakeholder Management

• Identifies, analyzes, plans, and manages stakeholder engagement throughout the project lifecycle. Added in PMBOK 5th Ed.

5. PMI Framework

The PMI Framework is the structured set of standards, processes, and guidelines provided by the Project Management Institute (PMI) for managing projects, codified primarily in the PMBOK Guide.

5.1 PMBOK Guide Overview

- PMBOK = Project Management Body of Knowledge
- Currently at 7th Edition (2021); exam-relevant editions: 5th and 6th.
- Describes 49 processes (6th Ed.) organized into 5 Process Groups and 10 Knowledge Areas.
- Is a standard, not a methodology — describes WHAT to do, not HOW to do it.

5.2 PMI Certifications

Certification	Target Audience	Requirement
PMP (Project Management Professional)	Experienced PMs	36–60 months experience + exam
CAPM (Certified Associate PM)	Entry-level PMs	23 hours PM education + exam
PgMP (Program Management Professional)	Program managers	PMP + 4 yrs program experience
PMI-ACP (Agile Certified Practitioner)	Agile teams	Agile experience + exam

5.3 PMI Code of Ethics

- Four core values mandated by PMI for all certified professionals:
  - Responsibility — taking ownership of decisions and actions
  - Respect — proper regard for oneself, others, and project resources
  - Fairness — making decisions and acting impartially
  - Honesty — understanding the truth and acting in a truthful manner

6. PMI Process Groups

PMI Process Groups are logical groupings of project management processes that work together to achieve a specific project objective; they are not sequential phases.

6.1 The Five Process Groups

6.x Initiating (2 processes)

• Authorize the project or phase. Key outputs: Project Charter, Stakeholder Register.

6.x Planning (24 processes)

• Establish scope, refine objectives, define course of action. Key output: Project Management Plan (PMP).

6.x Executing (10 processes)

• Complete work defined in the PMP. Key outputs: Deliverables, Work Performance Data, Change Requests.

6.x Monitoring & Controlling (12 processes)

• Track, review, and regulate progress. Key output: Work Performance Reports, Change Requests.

6.x Closing (1 process)

• Finalize all activities, close contracts, document lessons learned.

6.2 Process Group Interaction

• Initiating triggers Planning.
• Planning feeds into Executing.
• Monitoring & Controlling runs throughout Executing and feeds back change requests to Planning.
• Closing terminates after Executing is complete.
• Process Groups are NOT life cycle phases — multiple Process Groups can occur within a single phase.

6.3 Process Group vs Knowledge Area Matrix (PMBOK 6th Ed.)

Knowledge Area	Initiating	Planning	Executing	M&C	Closing
Integration	1	1	1	2	1
Scope	–	4	–	2	–
Schedule	–	6	–	1	–
Cost	–	3	–	1	–
Quality	–	1	1	1	–
Resources	–	2	4	1	–
Communications	–	1	1	1	–
Risk	–	6	1	1	–
Procurement	–	1	1	1	1

Stakeholders	1	1	1	1	–

7. Understanding Organizations

Organizational structure defines how activities are directed, coordinated, and supervised in pursuit of organizational goals, and directly impacts project manager authority.

7.1 Functional Organization

- Staff grouped by specialty (Engineering, Finance, Marketing, etc.).
- Project manager has little or no authority — projects coordinated by functional managers.
- Pros: clear career paths, deep technical expertise, efficient resource use.
- Cons: poor cross-functional coordination, weak project focus, silo mentality.

7.2 Projectized Organization

- Staff organized entirely around projects; project manager has full authority.
- Team members are co-located and dedicated to a single project.
- Pros: strong project focus, fast decision-making, clear accountability.
- Cons: resource duplication, idle resources between projects, no functional home for staff.

7.3 Matrix Organization

- Hybrid: staff has both functional manager and project manager.
- Three sub-types:

Type	PM Authority	Resource Control	PM Role
Weak Matrix	Low	Functional Manager	Coordinator
Balanced Matrix	Shared	Shared	Part-time PM
Strong Matrix	High	Project Manager	Full-time PM

- Cons: dual reporting creates confusion, potential for power conflicts.

7.4 Organizational Culture and its Effect on Projects

- Culture — shared values, norms, and practices within an organization.
- Culture influences: risk tolerance, communication style, decision-making speed, teamwork.
- Enterprise Environmental Factors (EEFs) — conditions not under project team control:
  - Organizational culture, structure, and governance
  - Infrastructure, existing resources, commercial databases
  - Government/industry standards, marketplace conditions
  - Stakeholder risk tolerances
- Organizational Process Assets (OPAs) — plans, processes, policies, knowledge bases:
  - Templates, lessons learned repositories, historical data
  - Organizational policies, procedures, and guidelines

8. Project Planning

Project planning is the process of defining scope, objectives, schedule, cost, quality, resources, and risk mitigation strategies to guide project execution.

8.1 Project Management Plan (PMP)

- Master document that integrates all subsidiary plans.
- Subsidiary Plans include:
  - Scope Management Plan
  - Requirements Management Plan
  - Schedule Management Plan
  - Cost Management Plan
  - Quality Management Plan
  - Resource Management Plan
  - Communications Management Plan
  - Risk Management Plan
  - Procurement Management Plan
  - Stakeholder Engagement Plan
- Baselines included: Scope Baseline, Schedule Baseline, Cost Baseline.

8.2 Scope Planning

- Steps:
  - 1. Collect Requirements — interviews, focus groups, surveys, prototyping
  - 2. Define Scope — produce Project Scope Statement
  - 3. Create WBS — decompose deliverables into manageable work packages
  - 4. Define Activities — further decompose work packages into schedule activities
- Project Scope Statement includes:
  - Product scope description
  - Deliverables list
  - Acceptance criteria
  - Project exclusions
  - Constraints and assumptions

8.3 Schedule Planning

- Process steps:
  - 1. Define Activities — decompose WBS work packages into activities
  - 2. Sequence Activities — identify dependencies (FS, SS, FF, SF)
  - 3. Estimate Activity Durations — using historical data, expert judgment, parametric
  - 4. Develop Schedule — CPM or PERT to produce project schedule
- Dependency Types:

Type	Code	Meaning
Finish-to-Start	FS	B cannot start until A finishes (most common)
Start-to-Start	SS	B cannot start until A starts

Finish-to-Finish	FF	B cannot finish until A finishes
Start-to-Finish	SF	B cannot finish until A starts (rare)

8.4 Cost Planning

- Cost Estimating Techniques:
  - Analogous Estimating — use actuals from similar past projects (low accuracy, low cost)
  - Parametric Estimating — statistical model using historical parameters (e.g., cost per LOC)
  - Bottom-up Estimating — estimate each work package and roll up (high accuracy, high cost)
  - Three-Point Estimating — (O + 4M + P) / 6 (handles uncertainty)
- Cost Baseline = approved budget used to measure and monitor cost performance
- Management Reserve — budget for unknown unknowns (outside cost baseline)
- Contingency Reserve — budget for known risks (inside cost baseline)

9. Project Evaluation

Project evaluation is the systematic assessment of a proposed or ongoing project to determine its viability, value, and alignment with organizational objectives.

9.1 Feasibility Study

- Conducted before project approval to assess whether the project is viable.
- Types of feasibility:
  - Technical Feasibility — can the technology support the solution?
  - Economic Feasibility — does the benefit justify the cost?
  - Operational Feasibility — will users accept and use the system?
  - Legal Feasibility — are there regulatory or contractual barriers?
  - Schedule Feasibility — can it be completed within required timeframes?

9.2 Business Case

- Documented justification for undertaking a project.
- Contents:
  - Business need or problem statement
  - Analysis of the situation (current state)
  - Recommended solution and alternatives considered
  - Expected benefits (tangible and intangible)
  - Cost estimates and financial projections
  - Risks and assumptions
  - Timeline

9.3 Financial Evaluation Techniques

- Net Present Value (NPV):
  - Sum of present values of all cash flows (inflows minus outflows)
  - NPV > 0: project adds value; NPV < 0: project destroys value
  - Formula: $NPV = \Sigma [CF_t / (1 + r)^t]$
- Return on Investment (ROI):
  - ROI = (Net Benefit / Cost) × 100%
  - Measures the efficiency of an investment
- Payback Period:
  - Time required to recover the initial investment
  - Shorter payback period = lower financial risk
- Internal Rate of Return (IRR):
  - Discount rate at which NPV = 0
  - Accept if IRR > cost of capital

9.4 Project Selection Methods

- Benefit Measurement Methods:
  - Comparative approach — rank projects against each other
  - Scoring models — weight criteria and score each project

  – Murder board — panel challenges project justification
- Constrained Optimization Methods:
  – Mathematical models used for large projects (linear programming, integer programming)
  – Used when there are resource constraints across multiple projects

10. Selection of an Appropriate Approach

Approach selection is the process of choosing between predictive (waterfall), adaptive (agile), or hybrid methodologies based on project characteristics, risk, and stakeholder needs.

10.1 Predictive (Waterfall) Approach

- Phases are sequential; previous phase must complete before next begins.
- Phases: Requirements → Design → Implementation → Testing → Deployment → Maintenance
- Best suited when:
  - Requirements are well-defined and stable
  - Technology is understood
  - Project is large, complex, and long-duration
  - Regulatory compliance documentation is required
- Pros: clear milestones, defined deliverables, easy to manage.
- Cons: inflexible to change, late delivery of working software, testing is late.

10.2 Adaptive (Agile) Approach

- Iterative and incremental; working software delivered in short cycles (sprints).
- Agile Manifesto (2001) — 4 values:
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan
- 12 Principles: early delivery, welcome change, sustainable pace, face-to-face communication, etc.
- Common frameworks: Scrum, Kanban, XP (Extreme Programming), SAFe.
- Best suited when: requirements are unclear/evolving, fast delivery needed, customer engagement is high.

10.3 Scrum Framework

- Roles: Product Owner, Scrum Master, Development Team.
- Artifacts: Product Backlog, Sprint Backlog, Increment.
- Events: Sprint (1–4 weeks), Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective.
- Definition of Done (DoD) — agreed criteria for a backlog item to be considered complete.

10.4 Hybrid Approach

- Combines elements of predictive and adaptive approaches.
- Example: Waterfall for requirements and architecture; Agile for development sprints.
- Used when some parts of the project are stable and others are uncertain.

10.5 Selection Criteria

Factor	Predictive	Agile
Requirements	Fixed, clear	Unclear, evolving

Team size	Any	Small (5–9 preferred)
Customer involvement	Low/milestone	Continuous
Regulatory	High (audit trail)	Low–Medium
Delivery	Single final release	Incremental releases
Risk tolerance	Low	Higher

11. Software Effort Estimation

Software effort estimation is the quantitative prediction of the cost and duration required to develop a software system, based on project size and complexity.

11.1 Lines of Code (LOC)

- Measure software size by counting source lines.
- Effort = productivity rate × LOC (e.g., 10 LOC/hour → 1000 LOC requires 100 hours).
- Issues:
  - Language-dependent (C requires more LOC than Python for same function)
  - Does not measure functionality or quality
  - Cannot be estimated early in the project

11.2 Function Point Analysis (FPA)

- Measures software size in terms of functionality delivered to users.
- Five function types:
  - External Inputs (EI) — data entering the system
  - External Outputs (EO) — data leaving the system
  - External Inquiries (EQ) — input-output combinations
  - Internal Logical Files (ILF) — data maintained within the system
  - External Interface Files (EIF) — data from other systems used as reference
- Each function type classified as Simple, Average, or Complex (weights assigned).
- Unadjusted Function Points (UFP) = sum of weighted counts.
- Adjusted FP = UFP × (0.65 + 0.01 × Σ complexity adjustment values)
- Language-independent; can be measured from requirements.

11.3 COCOMO Model

- Constructive Cost Model — algorithmic cost model by Barry Boehm (1981).
- Three levels:
  - Basic COCOMO: Effort = $a \times (KLOC)^b$
  - Intermediate COCOMO: adds 15 cost drivers (product, hardware, personnel, project attributes)
  - Detailed COCOMO: applies cost drivers at each phase
- Three project modes:

Mode	Description	a	b
Organic	Small team, well-understood problem	2.4	1.05
Semi-detached	Medium complexity and team	3.0	1.12
Embedded	Tight constraints, high complexity	3.6	1.20

11.4 COCOMO II

- Updated model (1995) for modern software development practices.
- Accounts for reuse, object-oriented design, and rapid development.
- Three sub-models:

– Application Composition — prototype phase (estimated in Object Points)
– Early Design — architecture phase (estimated in Function Points or KLOC)
– Post-Architecture — main development (most detailed, most accurate)

11.5 Expert Judgment (Delphi Technique)

- Multiple experts independently estimate effort.
- Estimates are aggregated and fed back anonymously.
- Process repeated until convergence.
- Used when historical data is insufficient.

11.6 Planning Poker (Agile)

- Team-based estimation using story points (relative effort).
- Fibonacci-like card deck: 1, 2, 3, 5, 8, 13, 21, 40, 100.
- Steps: read story → all vote simultaneously → discuss outliers → re-vote until consensus.

12. Activity Planning

Activity planning converts the Work Breakdown Structure (WBS) into a detailed schedule of activities with defined sequence, duration, and resource assignments.

12.1 Activity Definition

- Decompose WBS work packages into schedule activities.
- Activity Attributes: ID, name, description, predecessor/successor, resource requirements, constraints, assumptions, imposed dates.
- Milestone List: identifies key milestone events (duration = 0).

12.2 Activity Sequencing

- Identify and document dependencies between activities.
- Dependency types:
  - Mandatory (hard logic) — inherent in work nature (e.g., design before code)
  - Discretionary (soft logic) — best practices, preferred sequence
  - External — dependency on outside project (e.g., hardware delivery)
  - Internal — dependency within the project
- Leads and Lags:
  - Lead — overlapping activities (negative lag), e.g., start testing 2 days before coding ends
  - Lag — delay between activities, e.g., wait 3 days for concrete to dry

12.3 Critical Path Method (CPM) — Detailed

- Forward Pass (calculate ES and EF):
  - ES of first activity = 0
  - EF = ES + Duration
  - ES of successor = max EF of all predecessors
- Backward Pass (calculate LS and LF):
  - LF of last activity = EF of last activity
  - LS = LF – Duration
  - LF of predecessor = min LS of all successors
- Float = LS – ES (or LF – EF)
- Critical Path: all activities with Float = 0
- Total Float: amount of time activity can be delayed without delaying project end.
- Free Float: amount of time activity can be delayed without affecting ES of successor.

12.4 Schedule Compression Techniques

- Crashing:
  - Add resources to critical path activities to shorten duration
  - Increases cost; must always crash critical path activities only
  - Cost-slope = (Crash Cost – Normal Cost) / (Normal Duration – Crash Duration)
- Fast Tracking:
  - Perform activities in parallel that were originally sequential

  – Increases risk; rework may result
- Resource Leveling:
  – Adjust schedule to smooth resource demand
  – May extend the schedule (stretches non-critical activities)

13. Risk Management

Risk management is the systematic process of identifying, analyzing, and responding to project risks to maximize positive outcomes and minimize negative impacts.

13.1 Risk Terminology

- Risk — uncertain event or condition that, if it occurs, affects project objectives.
- Threat — negative risk (downside).
- Opportunity — positive risk (upside).
- Risk Probability — likelihood of the risk occurring (0–1 or %).
- Risk Impact — effect on objectives if risk occurs.
- Risk Exposure (Risk Score) = Probability × Impact.
- Residual Risk — risk remaining after responses are implemented.
- Secondary Risk — new risk created by implementing a risk response.

13.2 Risk Identification

- Techniques:
  - Brainstorming — team generates risks without judgment
  - Delphi Technique — structured expert opinion gathering
  - Interviews — stakeholder and expert interviews
  - Root Cause Analysis — identify root causes to find risks
  - SWOT Analysis — Strengths, Weaknesses, Opportunities, Threats
  - Checklist Analysis — use historical risk checklists from similar projects
  - Assumption Analysis — challenge project assumptions
- Output: Risk Register — living document listing all identified risks.

13.3 Qualitative Risk Analysis

- Prioritizes risks using subjective assessment of probability and impact.
- Probability and Impact Matrix:
  - Assign Low/Medium/High to probability and impact
  - Multiply P × I to get risk score
  - Classify risks as Red (high), Yellow (medium), Green (low)
- Used to filter which risks receive detailed quantitative analysis.

13.4 Quantitative Risk Analysis

- Numerically analyzes the effect of identified risks on project objectives.
- Techniques:
  - Expected Monetary Value (EMV) = Probability × Monetary Impact
  - Decision Tree Analysis — evaluates alternative decisions under uncertainty
  - Monte Carlo Simulation — runs thousands of possible scenarios to determine probability distribution of outcomes
  - Sensitivity Analysis (Tornado Diagram) — shows which risks have most impact on objectives

13.5 Risk Response Strategies

- For Threats (Negative Risks):
  - Avoid — eliminate the threat by changing the plan
  - Transfer — shift impact to a third party (insurance, outsourcing)
  - Mitigate — reduce probability or impact to acceptable level
  - Accept — acknowledge risk and take no action (active: contingency reserve; passive: document only)
- For Opportunities (Positive Risks):
  - Exploit — ensure opportunity occurs (assign best resources)
  - Share — allocate opportunity to a third party better able to capitalize
  - Enhance — increase probability or impact of opportunity
  - Accept — take advantage if opportunity arises, but don't actively pursue it

14. Evaluating Risks to the Schedule

Schedule risk evaluation quantifies the impact of uncertainties on project timeline using probabilistic techniques to determine likely completion dates.

14.1 PERT (Three-Point Estimation)

- Applies three-point estimation to activity durations.
- $t_e = (O + 4M + P) / 6$ (expected duration)
- Variance $\sigma^2 = [(P – O) / 6]^2$
- Project variance = sum of variances of critical path activities.
- Project standard deviation $\sigma = \sqrt{}$(sum of critical path variances)
- Probability calculations using Z-score: $Z = (T_d – \mu) / \sigma$
  - T_d = target date, μ = expected project completion date
  - Z-score maps to probability from standard normal distribution table

14.2 Schedule Risk Factors

- Resource availability — key person unavailability delays critical activities.
- Technical complexity — unfamiliar technology increases task duration uncertainty.
- External dependencies — vendor deliveries outside project control.
- Scope creep — uncontrolled additions extend schedule.
- Optimism bias — estimators systematically underestimate durations.

14.3 Schedule Reserve Analysis

- Schedule Reserve (Buffer) — time added to project schedule to reduce risk.
- Types:
  - Activity Buffer — added to individual activity duration estimates
  - Project Buffer — added at end of project schedule (Critical Chain PM)
  - Feeding Buffer — added at merge points where non-critical chains join critical chain
- Critical Chain Project Management (CCPM):
  - Resource dependencies explicitly included in schedule
  - Removes individual activity buffers; adds project and feeding buffers
  - Focus on protecting the project completion date, not individual task dates

15. Risk Control

Risk control (Monitor and Control Risks) is the ongoing process of implementing risk response plans, tracking identified risks, identifying new risks, and evaluating risk process effectiveness throughout the project.

15.1 Risk Monitoring Activities

- Risk Reassessment — regularly re-evaluate risks for changes in probability or impact.
- Risk Audits — examine and document effectiveness of risk response activities.
- Variance and Trend Analysis — compare actual to planned performance.
- Technical Performance Measurement — compare technical achievements to plan.
- Reserve Analysis — compare reserve consumption to the amount expected at that point.
- Risk Register Updates — close risks that are no longer applicable.

15.2 Risk Register Contents

- Risk ID and description
- Risk category (technical, external, organizational, PM-related)
- Probability and impact ratings
- Risk score (P × I)
- Risk owner (responsible for monitoring)
- Response strategy and specific actions
- Contingency plans and fallback plans
- Current status and trigger conditions

15.3 Risk Response Monitoring

- Trigger conditions (risk symptoms) — early warning signals that a risk is about to occur.
- Workarounds — unplanned responses to risks that occur without prior response plan.
- Change Requests generated from:
  - Corrective actions — bring project back in alignment with plan
  - Preventive actions — reduce probability of negative consequences
  - Defect repair — correct identified product defects

16. Configuration Management & Maintenance

Configuration Management (CM) is the process of identifying, tracking, controlling, and auditing changes to software items and their associated documentation throughout the software lifecycle.

16.1 Configuration Management Concepts

- Configuration Item (CI) — any software artifact placed under CM control:
  - Source code, executable files, documentation, test scripts, build scripts
  - Hardware descriptions, database schemas, configuration files
- Baseline — a formally reviewed and agreed-upon CI version that serves as basis for further development.
- Types of Baselines:
  - Functional Baseline — approved system requirements
  - Allocated Baseline — approved design specifications
  - Product Baseline — approved product configuration at end of development

16.2 CM Activities

- 1. Configuration Identification:
  - Identify and document physical/functional characteristics of CIs
  - Assign unique identifiers and version numbers
- 2. Configuration Control:
  - Evaluate, approve/reject, and implement changes to CIs
  - Change Control Board (CCB) — formal body that reviews change requests
  - Change Request → Impact Analysis → Approval/Rejection → Implementation → Verification
- 3. Configuration Status Accounting:
  - Maintain records of all CIs and their current status
  - Reports on baseline status, change request status, and version history
- 4. Configuration Audits:
  - Functional Configuration Audit (FCA) — verifies CI meets its requirements
  - Physical Configuration Audit (PCA) — verifies CI documentation matches actual product

16.3 Version Control Systems

System	Type	Key Feature
Git	Distributed	Every developer has full history; branching is cheap
SVN	Centralized	Single repository; simpler but single point of failure
Mercurial	Distributed	Similar to Git, used in some large organizations
Perforce	Centralized	Handles large binary files; used in game development

16.4 Branching Strategies

- Feature Branching — each new feature developed in dedicated branch; merged after review.
- Git Flow — main, develop, feature, release, and hotfix branches with defined merge rules.
- Trunk-Based Development — all developers integrate to trunk (main) at least daily.

16.5 Software Maintenance

- Modification of software after delivery to correct faults, improve performance, or adapt to new environment.
- Types (ISO/IEC 14764):
  - Corrective — fix bugs and defects
  - Adaptive — modify to work in changed environment (new OS, hardware)
  - Perfective — improve performance or maintainability
  - Preventive — prevent future faults (refactoring, code restructuring)
- Maintenance typically accounts for 60–80% of total software cost over its lifetime.

17. Environment for Configuration Control

The configuration control environment is the infrastructure, tools, processes, and organizational structures that enable controlled management of software configuration items across their lifecycle.

17.1 Software Configuration Management Environment (SCME)

- Components:
  - – Version Control Repository — stores all versions of CIs (e.g., Git repository)
  - – Build System — automated tools to compile, link, and package software (Make, Maven, Gradle)
  - – Issue/Change Tracking System — records defects and change requests (Jira, Bugzilla)
  - – Continuous Integration (CI) Server — automatically builds and tests on each commit (Jenkins, GitHub Actions)
  - – Artifact Repository — stores built artifacts, binaries, libraries (Nexus, Artifactory)

17.2 Change Control Process

- Step 1: Change Request Submission — documented request for modification.
- Step 2: Impact Analysis — assess effect on scope, schedule, cost, quality, risk.
- Step 3: CCB Review — approve, reject, or defer the change.
- Step 4: Implementation — code and test the approved change.
- Step 5: Verification — confirm change meets requirements.
- Step 6: Release — integrate and deploy the change.
- Step 7: Documentation Update — update affected documentation.

17.3 Continuous Integration / Continuous Delivery (CI/CD)

- CI — practice of merging all developer changes frequently (at least daily) and running automated tests.
- CD (Delivery) — automatically delivers code to a staging environment after CI succeeds.
- CD (Deployment) — automatically deploys to production after CD succeeds.
- Benefits:
  - – Early defect detection
  - – Reduced integration risk
  - – Faster release cycles
  - – Consistent build and test environment

18. Resource Allocation

Resource allocation is the process of identifying, assigning, and managing the assets needed to successfully complete a project, including human resources, equipment, materials, and budget.

18.1 Types of Project Resources

- Human Resources — project team members, specialists, contractors.
- Physical Resources — equipment, materials, facilities, infrastructure.
- Financial Resources — budget allocations.
- Time — often treated as a resource constraint.

18.2 Resource Estimation

- Bottom-up — estimate resources for each activity, then sum.
- Analogous — use resource data from similar past projects.
- Expert Judgment — rely on subject matter experts.
- Resource Breakdown Structure (RBS) — hierarchical list of resources by category.

18.3 Resource Histogram

- Bar chart showing resource demand over time.
- Horizontal axis = time; vertical axis = resource units (person-days, hours).
- Identifies resource over-allocation (demand exceeds availability) and under-utilization.

18.4 Resource Leveling

- Technique to resolve resource over-allocation by adjusting activity start/finish dates.
- Methods:
  - Delay activities with float — use available slack to push non-critical activities
  - Split activities — break activity into segments with pause between
  - Reassign resources — move resources from non-critical to over-allocated activities
- Effect: may extend project duration; maintains resource within availability limits.

18.5 Resource Smoothing

- Adjusts activities within their free and total float to minimize resource demand fluctuation.
- Does NOT extend project duration (unlike leveling).
- Less powerful than leveling — may not fully resolve over-allocations.

18.6 Team Management

- Tuckman's Model of Team Development:
  - Forming — team assembles; roles unclear; polite but unproductive
  - Storming — conflict over roles and approach; leader must manage conflicts
  - Norming — team establishes norms and cohesion; productivity increases
  - Performing — team is highly productive; minimal supervision needed

  – Adjourning — project ends; team disbands; lessons learned
- Motivation Theories:
  – Maslow's Hierarchy — physiological, safety, social, esteem, self-actualization
  – Herzberg's Two-Factor — hygiene factors (prevent dissatisfaction) vs. motivators (create satisfaction)
  – McGregor's X/Y — Theory X: workers are lazy; Theory Y: workers are self-motivated

19. Monitoring & Control

Project monitoring and control is the process of tracking, reviewing, and regulating project progress and performance against the project management plan, and initiating changes when necessary.

19.1 Earned Value Management (EVM)

- Quantitative method integrating scope, schedule, and cost to measure project performance.
- Three core values:
  - Planned Value (PV) — budgeted cost of work scheduled to be completed by a date
  - Earned Value (EV) — budgeted cost of work actually completed by that date
  - Actual Cost (AC) — actual cost incurred for work completed by that date
- Performance Indices:
  - Schedule Variance (SV) = EV – PV (SV < 0 = behind schedule)
  - Cost Variance (CV) = EV – AC (CV < 0 = over budget)
  - Schedule Performance Index (SPI) = EV / PV (SPI < 1.0 = behind schedule)
  - Cost Performance Index (CPI) = EV / AC (CPI < 1.0 = over budget)
- Forecasting:
  - Estimate at Completion (EAC) = BAC / CPI (if current CPI persists)
  - Estimate to Complete (ETC) = EAC – AC
  - To-Complete Performance Index (TCPI) = (BAC – EV) / (BAC – AC)
- BAC = Budget at Completion (total project budget)

19.2 Integrated Change Control

- All change requests must pass through a formal review before implementation.
- Change Control Board (CCB) — reviews and approves/rejects changes.
- Approved changes may affect: Project Management Plan, project documents, baselines.
- Change types:
  - Corrective Action — realign future performance with plan
  - Preventive Action — reduce probability of future negative variances
  - Defect Repair — fix identified defects

19.3 Project Reporting

- Work Performance Reports — compiled from Work Performance Information.
- Status Reports — current state (budget consumed, schedule variance, risks).
- Progress Reports — what has been accomplished in a period.
- Forecast Reports — predictions of future project status.
- Dashboard Reporting — visual, real-time display of key metrics (RAG status: Red/Amber/Green).

19.4 Quality Control

- Inspection — examining a work product to determine it conforms to standards.
- Control Charts — statistical charts tracking process output over time; identify special cause variation.
  - Upper Control Limit (UCL) and Lower Control Limit (LCL) = mean ± 3σ
  - Rule of Seven — 7 consecutive points on same side of mean = assignable cause

- Pareto Analysis — 80% of defects caused by 20% of root causes; prioritize the vital few.
- Cause-and-Effect Diagram (Ishikawa/Fishbone) — identifies root causes of defects.
- Statistical Sampling — inspect a representative sample rather than the entire population.

20. Review and Evaluation

Project review and evaluation is the systematic examination of project performance, deliverable quality, and process effectiveness at defined points in the project lifecycle.

20.1 Types of Reviews

- Phase Gate Reviews (Stage Gate):
  - Conducted at end of each project phase
  - Deliverables reviewed against acceptance criteria
  - Go/No-Go/Recycle decision made
- Sprint Reviews (Agile):
  - Team demonstrates completed work to product owner and stakeholders
  - Product backlog may be updated based on feedback
- Post-Implementation Review (PIR):
  - Conducted after deployment; assess whether project met business objectives
  - Compares actual benefits to projected benefits in business case
- Technical Reviews:
  - Walkthrough — author presents code/design to peers
  - Inspection — formal review with defined roles (moderator, reviewer, scribe)
  - Peer Review — informal examination by one or more peers

20.2 Lessons Learned

- Documented knowledge gained during a project that can be used on future projects.
- Collected throughout the project, not just at closure.
- Categories:
  - What went well — practices to be repeated
  - What went poorly — areas for improvement
  - Recommendations — specific process changes for future projects
- Stored in Lessons Learned Repository (OPA) for organizational benefit.

20.3 Project Closure

- Formal acceptance from the customer/sponsor that deliverables meet requirements.
- Administrative closure:
  - Archive project records
  - Release project team
  - Close procurement contracts
  - Update organizational process assets
  - Document lessons learned
- Premature closure — project ended before completion (cancelled, redirected).
  - Still requires formal closure process, lessons learned, and handover of partial deliverables

21. Challenges of Outsourcing in Project Management

Outsourcing in software PM is the contractual arrangement where an organization delegates software development or project management activities to an external vendor, often in a different geographic location.

21.1 Types of Outsourcing

Type	Definition	Example
Onshore	Vendor in same country	US firm hires US contractor
Nearshore	Vendor in neighboring/nearby country	US firm outsources to Mexico
Offshore	Vendor in distant country, often for cost savings	US firm outsources to India/China
Multisourcing	Using multiple vendors for different services	Cloud + dev + support from 3 vendors

21.2 Reasons for Outsourcing

- Cost reduction — lower labor costs in offshore locations.
- Access to specialized skills not available internally.
- Focus on core competencies — delegate non-core work.
- Flexibility — scale resources up/down quickly.
- Time zone advantage — 24/7 development cycle.

21.3 Challenges and Risks

- Communication Barriers:
  - Language differences and varying communication styles
  - Time zone differences reduce real-time collaboration windows
  - Cultural differences in directness, hierarchy, and conflict resolution
- Quality Control:
  - Difficulty monitoring work quality remotely
  - Different quality standards and testing practices
  - Hidden defects discovered late in integration
- Knowledge Transfer:
  - Loss of institutional knowledge when vendor changes
  - Documentation gaps impede future maintenance
- Security and IP Risks:
  - Exposure of proprietary code and data to external parties
  - Jurisdiction differences in IP law and data protection
- Vendor Lock-in:
  - Dependency on vendor-specific technology or processes
  - High switching costs if vendor relationship deteriorates
- Contract and Legal Risks:
  - Ambiguous SLAs and performance metrics
  - Dispute resolution across jurisdictions
  - Regulatory compliance in multiple countries

21.4 Contract Types in Outsourcing

Contract Type	Description	Risk Allocation
Fixed Price (FP)	Agreed total price; vendor delivers	Vendor bears cost risk
Time & Material (T&M)	Pay per hour/unit of work	Buyer bears cost risk
Cost Plus Fixed Fee	Reimburse costs + fixed fee	Buyer bears most risk
Cost Plus Incentive	Reimburse costs + incentive for targets	Shared risk

21.5 Managing Outsourced Projects

- Service Level Agreements (SLAs):
  - Define measurable performance targets (defect rate, response time, delivery schedule)
  - Include penalties for non-compliance and bonuses for over-performance
- Governance:
  - Establish a Vendor Management Office (VMO) for oversight
  - Regular status reviews, steering committees, escalation paths
- Communication Plan:
  - Define meeting cadence across time zones
  - Establish shared collaboration tools (Confluence, Jira, Slack)
- Knowledge Management:
  - Mandatory documentation standards
  - Knowledge transfer sessions before vendor exit
- Exit Strategy:
  - Plan for vendor transition from project start
  - Retain critical IP in-house

Quick Reference: Key Formulas & Definitions

EVM Formulas

Metric	Formula	Interpretation
SV (Schedule Variance)	EV – PV	< 0: behind; > 0: ahead
CV (Cost Variance)	EV – AC	< 0: over budget; > 0: under budget
SPI	EV / PV	< 1: behind; > 1: ahead
CPI	EV / AC	< 1: over budget; > 1: under budget
EAC	BAC / CPI	Revised total project cost estimate
ETC	EAC – AC	Remaining work cost estimate
TCPI	(BAC–EV)/(BAC–AC)	Efficiency needed to finish on budget

Estimation Formulas

Formula	Name	Use
t_e = (O+4M+P)/6	PERT expected time	Activity duration estimation
σ<super>2</super> = [(P–O)/6]<super>2</super>	PERT variance	Schedule risk calculation
EMV = P × I	Expected Monetary Value	Quantitative risk analysis
ROI = (Benefit–Cost)/Cost × 100	Return on Investment	Project selection
NPV = Σ CF_t/(1+r)<super>t</super>	Net Present Value	Financial feasibility

CPM Formulas

- EF = ES + Duration
- LS = LF – Duration
- Float = LS – ES = LF – EF
- Critical Path: Float = 0 on all activities

End of Notes — Software Project Management